UNCERTAINTY QUANTIFICATION OF MACHINE LEARNED DENSITY FUNCTIONALS

A Dissertation
Presented to
The Academic Faculty

By

Karan N. Shah

In Partial Fulfillment
of the Requirements for the Degree
Bachelors of Science in the
School of Computer Science

Georgia Institute of Technology

May 2018
(Updated May 2019)

UNCERTAINTY QUANTIFICATION OF MACHINE LEARNED DENSITY FUNCTIONALS

Approved by:

Dr. Andrew Medford, Advisor
School of Chemical & Biochemical Engineering
Georgia Institute of Technology

Dr. Richard Vuduc
School of Computational Science & Engineering
Georgia Institute of Technology

Dr. David White
School of Computer Science
Georgia Institute of Technology

Date Approved: May 03, 2018

TABLE OF CONTENTS

List of Tables iii

List of Figures iv

Chapter 1: Introduction 1

Chapter 2: Background 3

Chapter 3: Computational Methods 6

3.1 Data Generation 6

3.2 Descriptors 6

3.3 Subsampling 7

3.4 Model Training 7

3.5 Neural Networks 8

3.6 Bootstrap Aggregation 9

3.7 Prediction 9

3.8 Tools 11

Chapter 4: Results and Discussion 12

4.1 LDA VWN Formulation 12

4.2 Neural Network Ensemble 13

4.2.1 Predicted Energy Density 14

4.2.2 Uncertainty Quantification 21

Chapter 5: Conclusion 25

References 30

LIST OF TABLES

2.1 Computational Complexity of DFT functionals 4

3.1 Neural Network Architecture 8

4.1 Sum of Absolute Energy Error per system (eV) 16

4.2 Mean and Max Prediction Errors with Uncertainty ($\frac{eV}{\mathring{A}^3}$) 19

4.3 Percentage of points with uncertainty error within uncertainty $n\sigma$ 22

LIST OF FIGURES

3.1 Neural Network Architecture 8

3.2 Training Workflow: The ensemble of Neural Networks is trained on bootstraped datasets consisting of residuals generated from the parameterized LDA function. . . 10

3.3 Prediction Workflow 10

4.1 $\epsilon_{xc-B3LYP}$ vs ρ 12

4.2 ϵ_{xc-res} vs ρ 14

4.3 ϵ_{pred}vs ρ and $\epsilon_{pred-NN}$ vs $\epsilon_{pred-VWN}$ 15

4.4 SAE per system 17

4.5 SAE per system, comparison between LDA-VWN and LDA-NN 18

4.6 Mean Prediction Error per system 20

4.7 Max Prediction Error per system 20

4.8 ϵ_{pred}vs ρ and $\epsilon_{pred-NN}$ vs $\epsilon_{pred-VWN}$ 21

4.9 Percentage of points with uncertainty error within uncertainty $n\sigma$ 23

4.10 SAE and Uncertainty for varying ensemble size, for C2H6 24

SUMMARY

Density Functional Theory is one of the most popular and successful methods for quantum mechanical simulations of matter because of its relatively low computational costs. While it is formally exact, approximations of exchange correlation (XC) functionals have to be made. These calculations are highly time consuming and scale poorly with system size. The prospect of combining computer vision and deep learning is a fundamentally new approach to designing these XC functionals. This approach combines the intuitive power of physical insight with the flexibility of machine learning and high-quality training data in order to develop new routes to approximating exchange-correlation energies. One challenge with machine-learned functionals is that their reliability is difficult to assess. Unlike physically-derived functionals, machine-learned functionals are accurate only in regions near training data. Developing strategies to quantify the uncertainty of machine-learned functionals will improve reliability and enable new strategies for generating training data. In this work, a parameterized function is first fit on the data and the resulting residuals are used for bootstrap aggregating via an ensemble of neural networks. This two-stage method provides robust uncertainty quantification on the predicted XC energies and can be automated for many systems without significant manual intervention.

CHAPTER 1
INTRODUCTION

The electronic structure of a many-body system is the state of motion of electrons in an electrostatic field created by stationary nuclei [1]. The investigation of electronic structure of many body systems has a plethora of applications, from battery development [2] to determination of the structure of planetary interiors [3]. There are two classes of methods to determine electronic structure: Wavefunction Theories (WFT) and Density Functional Theory (DFT) [4, 5]. DFT is advantageous to WFT because of its simple formalism and lower computational costs. However, the modeling of electronic structure using DFT currently scales poorly with system size and is still too expensive computationally for complex systems. While semi-empirical approximations to DFT result in a reduction in computational time versus *ab initio* DFT, creating such approximations involves significant manual intervention and is highly inefficient for high-throughput electronic structure screening calculations.

Computational material design based on machine learning techniques is a rapidly growing area in materials science [6]. Recent advances in computational power and deep learning algorithms enable us to carry out DFT calculations for a large number of compounds and crystal structures systematically. It is expected that machine learning techniques will be effective in finding a unique mapping between the electron density and the exchange-correlation (XC) potential of a system.

However, machine learning algorithms are only as good as the data they are trained on. A significant portion of this research was dedicated to the study of the sensitivity of predictions with respect to the training data and to uncertainty quantification of machine learned functionals. This was achieved through statistical techniques such as bootstrap aggregation and boosting. In this thesis the data extracted from DFT calculations of small molecule systems using the B3LYP functional is used to construct machine-learned XC functionals. These functionals use convolutions of the electron density to "fingerprint" electronic environments, and neural networks are used to con-

nect the fingerprints to the local XC energy. The key contribution of this thesis is the application of bootstrap aggregation to estimate the uncertainty associated with these machine-learned functionals. These ensembles are used to provide error bars for the machine-learned prediction, and the reliability of this approach is assessed by comparing to training and testing data. The findings provide insight into the design of robust machine-learning XC functionals.

Outline of thesis

Chapter 2 provides a brief background on DFT and machine learning methods. Computational details such as the data generation process and machine learning workflow are discussed in Chapter 3. Chapter 4 is devoted to results and related discussion. Concluding remarks are presented in Chapter 5.

CHAPTER 2

BACKGROUND

In 1964, Honenberg and Kohn [4] proved that the ground-state properties of many-electron systems can be uniquely determined by an electron density that depends on only 3 spatial coordinates. For a system of electrons, the ground state energy is the minimum value of a unique functional of the charge density $n(r)$ in the system. This is an exact result. However, to determine the exact energy, the formally exact but unknown exchange-correlation functional $E_{XC}[n(r)]$ is required. The first method to approximate $E_{XC}[n(r)]$, known as Local Density Approximation (LDA), was given by Kohn and Sham [5] in 1965. The search for a physically derived functional has been ongoing for over 50 years, but thus far no universal and systematically improvable approximation has been discovered [7].

The formalism of DFT proves that there is a unique functional mapping between the electron density and the exchange-correlation potential. It has been shown that approximations become more accurate as more non-local information is included [3]. The LDA [8] functional estimates the XC energy based only on the electron density at a given point, while the generalized gradient approximation (GGA) [9, 10] family of functionals includes the gradient, and meta-GGA [11] functionals include the kinetic energy density. As more information is included the accuracy generally increases, although the improvements are not systematic [12, 13]. "Jacob's ladder" [11] is a popular analogy to this tradeoff between generalizability and accuracy of an approximation as more non-local information is included. Increasing the amount of non-local information also increases the computational complexity of the approximation [14].

The computational complexities of various DFT functionals are:

Table 2.1: Computational Complexity of DFT functionals

Functional Family	Computational Complexity [15]
Orbital-free functionals (LDA, GGA, mGGA etc.)	$O(N^3)$
Hybrid Functionals(B3LYP, PBE0,HSE etc.)	$O(N^4)$
Double Hybrid Functionals (QIDH etc.)	$O(N^5)$

Neural networks [16, 17] are a promising approach to regression due to the "universal approximation theorem" [18] that shows that they are flexible enough to fit any function. Deep learning [19] has been widely applied to problems in different domains including, but not limited to, image classification [20], linguistics [21, 22], biomedical engineering [23] and cosmology [24, 25]. However, neural networks also have some challenges in determining hyperparameters and training routines that avoid overfitting. Recent advances in neural network architecture, such as dropout layers [26] are relatively robust to overfitting and are easy to implement [27]. Another approach to reduce overfitting and provide uncertainty bounds is ensemble learning [28]. Multiple learners are trained on different subsets of training data and their combined results are used to generate predictions. This has been shown to reduce the need for hyperparameter tuning and overfitting [29], but with increasing computation costs.

Surrogate density functionals can be created by training machine learning models on a large amount of high-quality data generated using computationally expensive simulations with a complex model space. The advantage of these machine learned approximations is that they can be generalized across different systems without the knowledge of underlying physics. Once the models are trained, inferring ϵ_{XC} is relatively cheaper computational cost wise. The accuracy of these models can also be increased by adding more training data. The key challenges for designing machine learned approximations are uncertainty quantification and preventing overfitting. Rupp [30] provides a succinct survey of contemporary machine learning techniques for quantum mechanical problems. However, due to the high dimensional space in which most molecules interact, QM

simulations are computationally very expensive. Mills et al [31] develop a transferable machine learning approach to infer the properties of a bound electron. They used 256×256 grid "images" of potentials for different systems as training data. Energies were used as labels in training data. The neural network served as an accurate mapping between energies and potentials, performing faster than the finite-difference method used to produce the training data. While promising, this system is too sensitive to training data and breaks down for predictions of more complex systems. Balabin and Lomakina [32] et al train a neural network on 208 different molecules and test it by applying BLYP, B3LYP, and BMK density functionals. Robust uncertainty quantification is needed to validate predictions against real world systems. Peterson et al [33] recently showed that this can be done through bootstrap aggregation(bagging) [34] with neural network ensembles. This technique uses multiple weak neural networks on subsamples of the training data to estimate uncertainty. Another advantage of bagging is that it reduces overfitting. Prior efforts have focused on quantifying uncertainty for machine-learned atomistic force fields [33].

This thesis explores the use of neural network ensembles towards uncertainty quantification of machine-learned density functionals. Neural networks are used as surrogate functionals between a simple density based descriptor space and exchange correlation energy. Uncertainty quantification is done through bootstrap aggregation.

CHAPTER 3
COMPUTATIONAL METHODS

The goal of this project was to train neural networks as surrogate functionals that map non-empirical physical information (LDA and GGA) to energy density calculated using higher order functionals such as B3LYP [35] .

3.1 Data Generation

The Psi4 package was used to generate large high quality datasets for 15 systems: C_2H_2, C_2H_4, C_2H_6, CH_3OH, CH_4, CO, CO_2, H_2, H_2O, HCN, HNC, N_2, N_2O, NH_3, O_3. These molecules contain 4 common atom types (C, H, O, N) and a diverse range of single, double, and triple bonds between them. The geometries of molecules were taken from computational chemistry comparison and benchmark database (CCCBDB) maintained by NIST [36] and are static for all calculations. Single point calculations were done on small molecule systems on B3LYP/aug-cc-pvtz level [12] with very tight convergence criteria (10^{-11} for both density and energy density). After convergence, the electronic density $\rho(r)$ and the exchange correlation energy density ϵ_{xc} of the systems were projected onto a 3D uniform finite-difference grid. Each grid was of volume $10\mathring{A}^3$ with the molecule located at the center and with a grid spacing of 0.02 Å. Thus, each system grid consisted of $500^3 = 125,000,000$ data points. The energy density $\epsilon(\rho)_{xc-B3LYP}$ at each point was calculated using B3LYP functional.

3.2 Descriptors

The electronic density $\rho(r)$ at each point in the grid were used to train the model, with the target variable being the B3LYP exchange correlation energy density $\epsilon(\rho)_{xc-B3LYP}$. The density gradient was computed numerically using finite-difference stencils [37]. The Scipy Fast Fourier Transform

convolution was used because of its $O(NLog(N))$ scaling as opposed to the $O(N^3)$ scaling of vanilla convolution [38].

3.3 Subsampling

Given the enormity of the dataset (10^8 datapoints per system), data was subsampled to create smaller but representative subsets which could be used to train neural networks in a feasible time frame. Simple uniform subsampling was not ideal because the data is highly repetitive, and most points are effectively vacuum, which would cause loss of information, especially through omission of points in the core region of the highest density. To counter this, an iterative subsampling algorithm based on kD-trees was used. After subsampling, 750,000 datapoints (out of 125,000,000 datapoints) per system were used to train the model.

3.4 Model Training

After the creation of subsampled dataset consisting of the extracted descriptors and the target, the data was fitted to an optimized Vosko-Wilk-Nusair LDA model (VWN) [39]. The VWN model was used to roughly fit the data in the bulk part and an ensemble of neural networks were trained on the resulting residuals. The VWN model has 6 parameters($C_1, \gamma, \alpha_1, \beta_1, \beta_2, \beta_3, \beta_4$) that were optimized based on molecular data. To improve the fit, the parameters were further tuned with the subsampled dataset in addition to 1,000,000 randomly subsampled data points. Instead of training a neural net directly on $\epsilon(\rho)_{xc-B3LYP}$ values, the residual energy density values

$$\epsilon(\rho)_{xc-res} = \epsilon(\rho)_{xc-B3LYP} - \epsilon(\rho)_{xc-VWN} \tag{3.1}$$

were used, where $\epsilon(\rho)_{xc-VWN}$ was calculated using the analytical VWN model.

3.5 Neural Networks

There are many design decisions are involved in creating neural networks. Multiple neural network architectures, including different layer sizes and activation functions like ReLu [40] were tested but only the results from the following architecture are presented for the sake of brevity. Neural networks consisting of an input layer, 2 fully connected layers of 20 nodes each and an output layer were used. The hidden layers had $tanh$ activation functions [41] the output layer had linear activation function. The input layer was of size 1 for LDA (using only $\rho(r)$). The optimizer used was ADAM [42] with a learning rate of 0.00001. The tolerance for loss was set to 10^{-7}, and training was force stopped if the loss did not go below that limit after 500 epochs. The neural network was trained on input $\rho(r)$ mapped to output $\epsilon(\rho)_{xc-res}$.

Table 3.1: Neural Network Architecture

Layer(type)	Output Shape	No. of Params.	Activation Fn
fc1 (Dense)	20	40	$tanh$
fc2 (Dense)	20	420	$tanh$
Output (Dense)	1	21	Linear

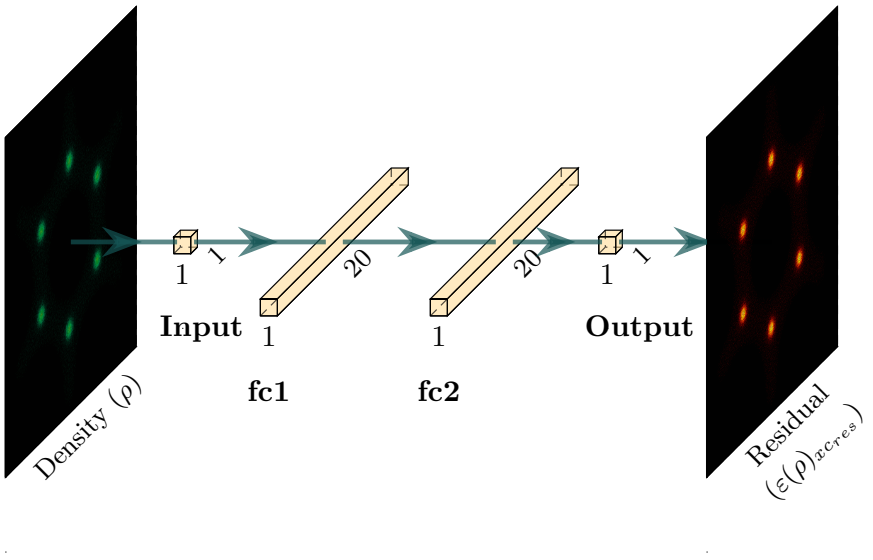


Figure 3.1: Neural Network Architecture

3.6 Bootstrap Aggregation

Bootstrapping refers to the act of creating subsets of the data by resampling from the original data distribution. This is done using random sampling with replacement., and therefore helps in reducing variance. Bagging increases the stability and decreases overfitting and variance when used for neural net based regression [34]. It is expected that each such bootstrapped set contains the fraction $(1 - \frac{1}{e})$ of the unique data points in the training set, with the rest being duplicates [43].Each subset of data is used to train a different model. When used for regression, the results of each model are used to calculate summary statistics which serve as the aggregated prediction, with the benefit of uncertainty quantification. In this project, 10 subsets with size equal to the training dataset were created by randomly sampling with replacement from the training set. A neural network was trained on each of the subsets.

3.7 Prediction

The whole ensemble is used to predict the residual for each data point in the test set, giving multiple predictions for each point. The summary statistics of predictions per data point were used as a measure of uncertainty.

Different error metrics were used to measure model performance. First, the local energy error at each point was calculated. This measure served to quantify how the models performed in different regions of a system. The system level energy is given by

$$E_{xc}(\rho(\vec{x})) = \int_{R^3} \epsilon_{xc}[\rho(\vec{x})](\vec{x}) \mathrm{d}^3\vec{x} \tag{3.2}$$

This would indicate models where the overall prediction is consistent but the local energy errors in different regions canceled each other out.

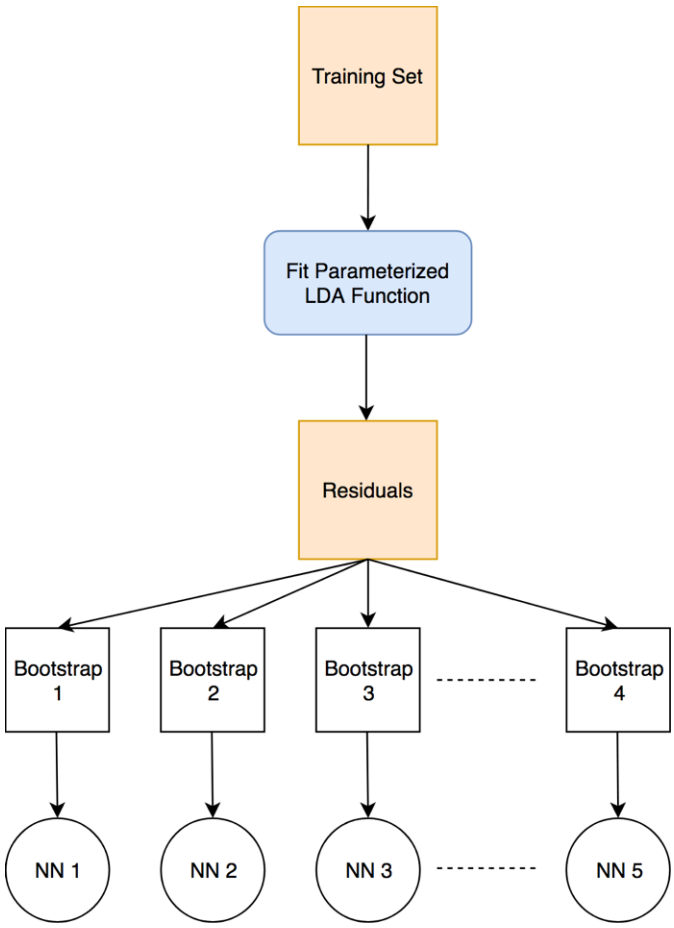


Figure 3.2: Training Workflow: The ensemble of Neural Networks is trained on bootstraped datasets consisting of residuals generated from the parameterized LDA function.

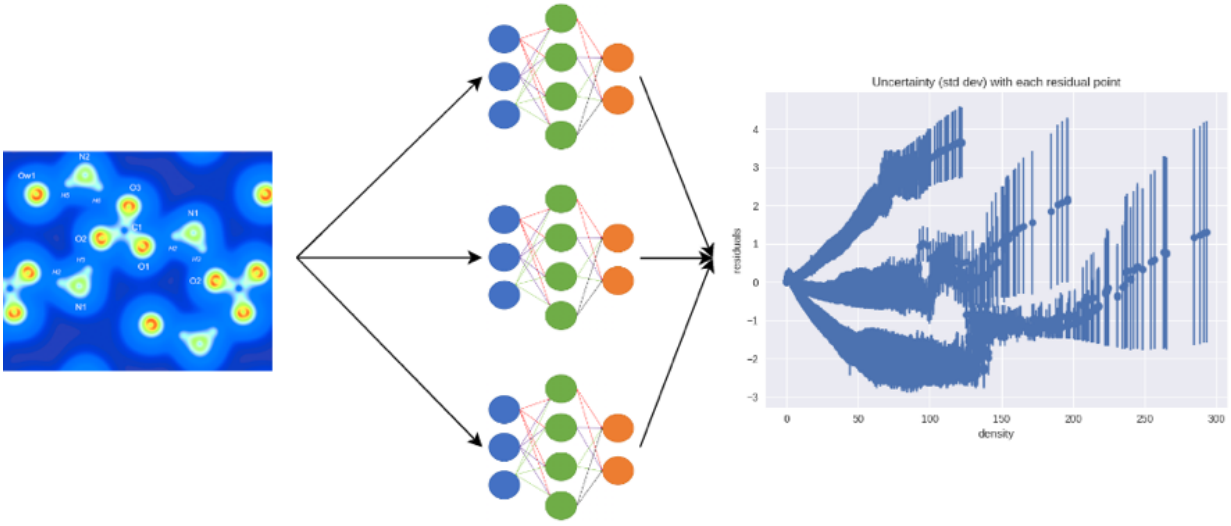


Figure 3.3: Prediction Workflow

3.8 Tools

Python was the primary programming language for this project. The two main libraries used were Psi4 [44] and Keras [45] with a Tensorflow backend [46]. Psi4 is an open-source suite of quantum chemistry programs designed for efficient, high-accuracy simulations of a variety of molecular properties. Keras is an API for rapid experimentation and prototyping of neural networks using Tensorflow. The majority of computation was done on PACE [47].

CHAPTER 4

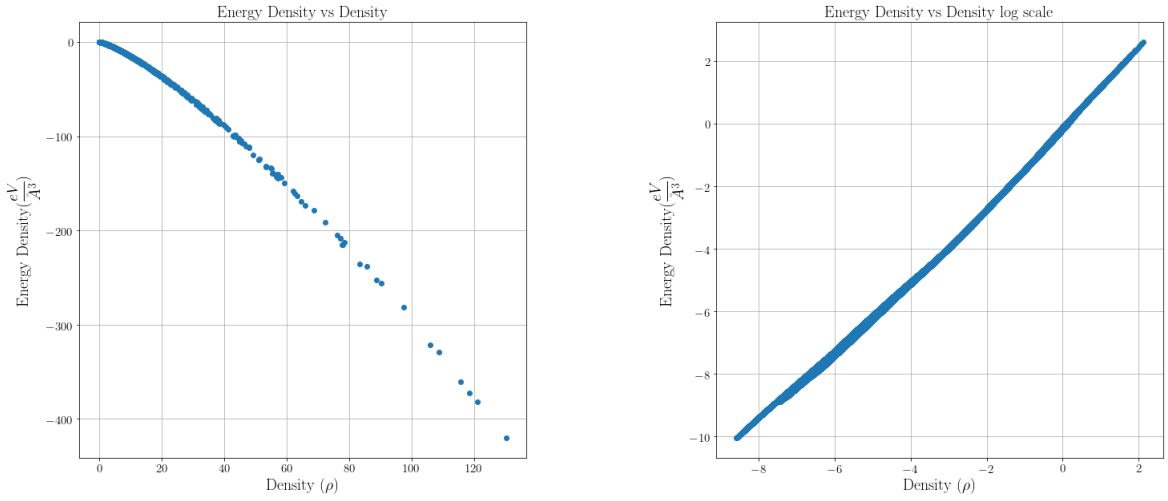
RESULTS AND DISCUSSION

This section is presented in two parts. In Section 4.1, the need for LDA residual fitting through VWN parametrization is discussed. In Section 4.2, the results using neural network ensembles are presented along with discussion about uncertainty quantification.

All scatter plots were generated using 2000000 subsampled data points unless otherwise specified. All summary statistics plots were generated using the entire dataset (12500000 points).

4.1 LDA VWN Formulation

While a vast majority of $\epsilon_{xc-B3LYP}$ values are in the range $10^{-6.5}$ to 10^{0} $(\frac{eV}{\mathring{A}^3})$, the range of all values, including high energy core regions, spans from -400 $(\frac{eV}{\mathring{A}^3})$to 0 $(\frac{eV}{\mathring{A}^3})$(Fig 4.1(a)) and 12 orders of magnitudes(Fig 4.1(b)). This is problematic for neural networks as small relative errors in the high energy regions can induce a large error in the systemwide energy predictions [48, 37].



(a) ϵ_{xc} vs ρ (b) log(-ϵ_{xc}) vs log(ρ)

Figure 4.1: $\epsilon_{xc-B3LYP}$ vs ρ

Physical models such as VWN can approximate the energy density across this vast range [39, 49]. The VWN parameterization of the LDA model uses an analytical function that reproduces the behavior of the homogeneous electron gas (HEG) [39]

$$\begin{aligned} e_{XC,VWN} &= e_{X,VWN} + e_{C,VWN} \\ &= \rho\frac{C_1}{r_s} + \rho G\left(r_s, \gamma, \alpha_1, \beta_1, \beta_2, \beta_3, \beta_4\right) \end{aligned} \tag{4.1}$$

where $C_1, \gamma, \alpha_1, \beta_1, \beta_2, \beta_3, \beta_4$ are the parameters, r_s is the Wigner-Seitz radius [50] defined as:

$$r_s = (3/4\pi\rho)^{1/3} \tag{4.2}$$

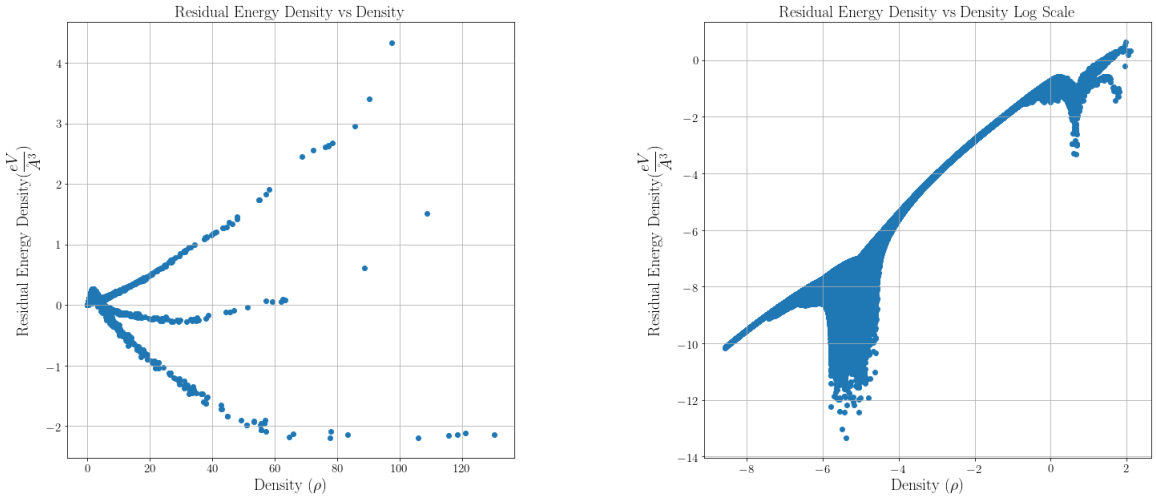
G is defined as:

$$\begin{aligned} G\left(r_s, \gamma, \alpha_1, \beta_1, \beta_2, \beta_3, \beta_4\right) &= -2\gamma\left(1 + \alpha_1 r_s\right) \\ \times \ln\left\{1 + \frac{1}{2\gamma r_s^{1/2}\left(\beta_1 + r_s^{1/2}\left(\beta_2 + r_s^{1/2}\left(\beta_3 + r_s^{1/2}\beta_4\right)\right)\right)}\right\} \end{aligned} \tag{4.3}$$

The range of the resulting residuals is significantly reduced to about -2 $(\frac{eV}{\mathring{A}^3})$–4 $(\frac{eV}{\mathring{A}^3})$(Fig 4.2(a)) with a majority of points between spanning 6 orders of magintude(Fig 4.2(b)). This lower variance in values allows the neural networks to learn the ρ -ϵ_{xc} mapping with better accuracy [51].

4.2 Neural Network Ensemble

The bootstrap aggregation process is used to generate a surrogate LDA functional LDA-NN. The performance of LDA-NN in prediction and uncertainity quantification of energy density ϵ_{xc} is presented in this section.



(a) ϵ_{xc} vs ρ

(b) log(-ϵ_{xc}) vs log(ρ)

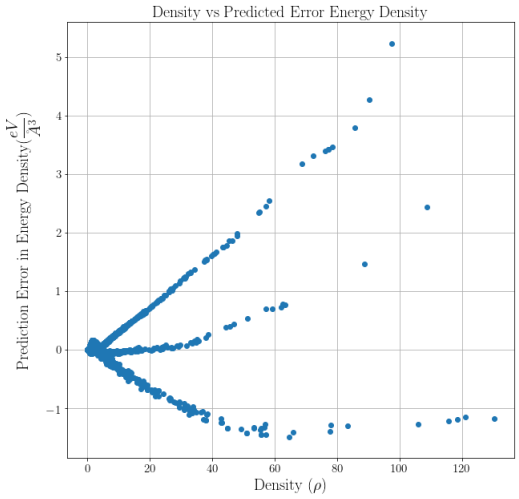
Figure 4.2: ϵ_{xc-res} vs ρ

4.2.1 Predicted Energy Density

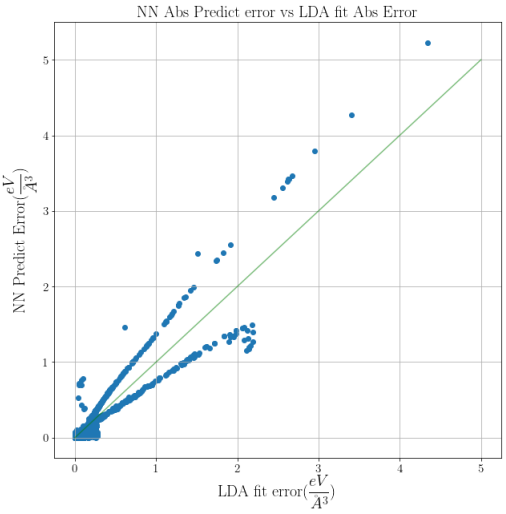
The local energy density prediction error is defined as

$$\epsilon_{pred}(\rho) = \varepsilon_{pred}(\rho) - \varepsilon_{xc-B3LYP}(\rho) \tag{4.4}$$

Figure 4.3 (a) shows the distribution of NN prediction errors with respect to ρ. The three prongs of outliers correspond to the high energy regions in different systems. The vast majority of the points lie in the low error range but this is not apparent in the scatter plot. When compared to VWN-LDA error, NN-LDA error (Fig 4.3 (b) lower for majority of the points and is evenly distributed around the identity line for high energy regions. This suggests that the ensemble system is minimizing the overall error with respect to residuals by overestimating and underestimating roughly half the outliers.



(a) Error in NN predictions vs Density



(b) Error in VWN predictions vs Error in NN predictions

Figure 4.3: ϵ_{pred}vs ρ and $\epsilon_{pred-NN}$ vs $\epsilon_{pred-VWN}$

Sum of Absolute Energy Error

The sum of absolute energy errors (SAE ε_{abs}) approximates the difference between true energy E_{xc} and predicted energy $E_{xc-pred}$. It is given by

$$\varepsilon_{\text{abs}} = \sum_i \left|\epsilon_{xc-B3LYP}\left(\vec{x}_i\right) - \epsilon_{xc-pred}\left(\vec{x}_i\right)\right| \times h^3 \tag{4.5}$$

It provides an upper bound for system level errors as there is no cancellation of errors.

Table 4.1: Sum of Absolute Energy Error per system (eV)

System	LDA-VWN	LDA NN	LDA NN Uncertainty
H2	0.34	0.20	0.18
CH4	1.53	0.27	0.21
NH3	1.48	0.29	0.21
H2O	1.48	0.43	0.20
C2H2	1.97	0.33	0.22
HCN	1.92	0.35	0.22
N2	1.87	0.37	0.22
C2H4	2.35	0.34	0.23
HNC	1.92	0.35	0.22
C2H6	2.73	0.35	0.25
CO	1.92	0.50	0.21
CH3OH	2.67	0.51	0.23
N2O	3.02	0.61	0.24
CO2	3.08	0.74	0.24
O3	3.40	0.86	0.24

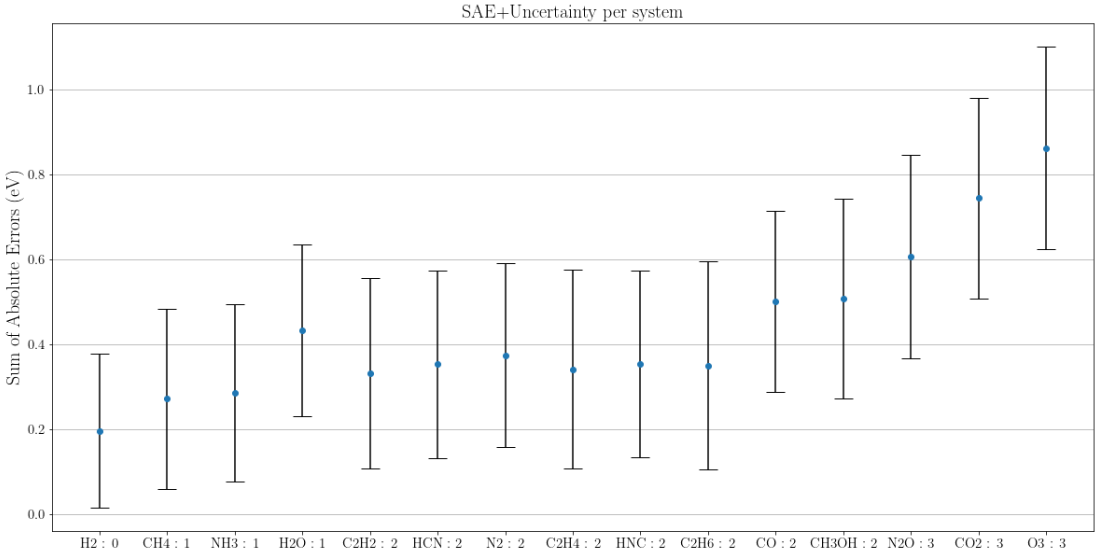


Figure 4.4: SAE per system

Fig 4.6 shows the SAE per system. The number of heavy atoms is listed per system. It is observed that the SAE increases as the number of heavy atoms in a system increases. The uncertainty in estimating SAE increases slightly as the number of heavy atoms increases and is roughly constant for systems with the same no. of heavy atoms (Table 4.2).

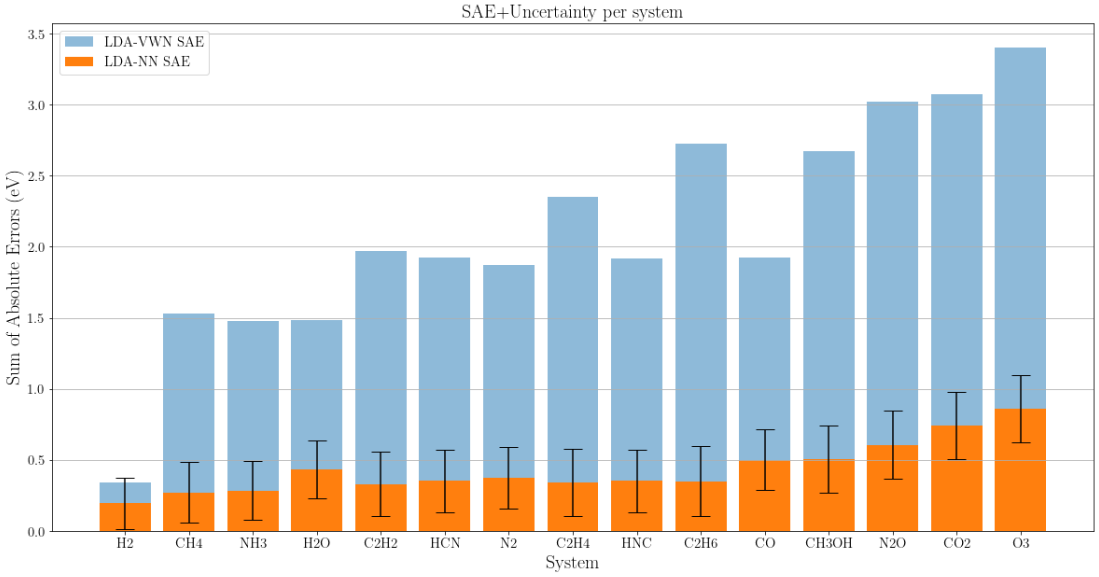


Figure 4.5: SAE per system, comparison between LDA-VWN and LDA-NN

The LDA-NN SAE is significantly less than LDA-VWN SAE. This is because VWN is less effective at predicting the energy density at regions where chemical bonding occurs, where there are an appreciable number of data points (29.5%). However, the SAE of LDA-NN, even within uncertainty bounds is greater than the SAE desired for chemical accuracy. This arises from the fact that ϵ_{xc} is not injective with ρ near the high energy core regions. The results can be improved by incorporating non-local information.

Max and Mean Prediction Error

The sum of prediction error is defined as

$$\epsilon_{sum} = \sum_i \left(\epsilon_{xc-B3LYP}\left(\rho_i\right) - \epsilon_{pred_{xc-pred}}\left(\rho_i\right) \right) \times h^3 \tag{4.6}$$

The mean prediction error is given by

$$\epsilon_{MPE} = \frac{\epsilon_{sum}}{n} \tag{4.7}$$

for a system of n points.

Similarly, the max prediction error is

$$\epsilon_{MaxPE} = Max(\epsilon_{pred}(\rho)) \tag{4.8}$$

Table 4.2: Mean and Max Prediction Errors with Uncertainty $(\frac{eV}{\mathring{A}^3})$

System	MPE	Mean Uncertainty	MaxPE	Max Uncertainty
H2	1.49e-04	1.81e-04	0.02	2.16e-03
CH4	1.31e-04	2.13e-04	17.69	3.28e-02
NH3	1.54e-04	2.09e-04	16.41	4.34e-02
H2O	2.20e-04	2.03e-04	9.72	4.61e-02
C2H2	1.47e-04	2.24e-04	17.70	3.30e-02
HCN	1.77e-04	2.20e-04	24.75	4.37e-02
N2	1.95e-04	2.16e-04	23.11	4.37e-02
C2H4	1.38e-04	2.34e-04	5.15	3.10e-02
HNC	1.80e-04	2.20e-04	14.07	4.32e-02
C2H6	1.18e-04	2.45e-04	7.99	3.18e-02
CO	2.39e-04	2.14e-04	17.73	4.59e-02
CH3OH	2.00e-04	2.35e-04	5.28	4.58e-02
N2O	2.70e-04	2.39e-04	27.31	4.61e-02
CO2	3.23e-04	2.36e-04	17.74	4.61e-02
O3	3.25e-04	2.39e-04	1.60	4.61e-02

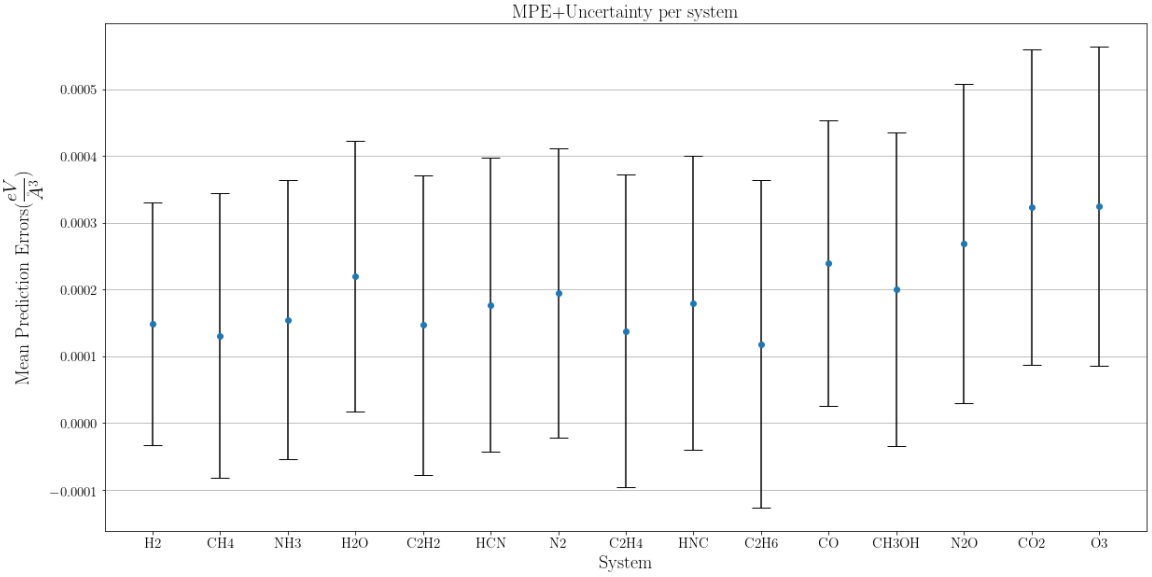


Figure 4.6: Mean Prediction Error per system

Fig 4.6 shows the systemwide MPE. It follows similar trends as SAE but is significantly lower because of cancellation of errors. The uncertainty bounds do not vary significantly by system.

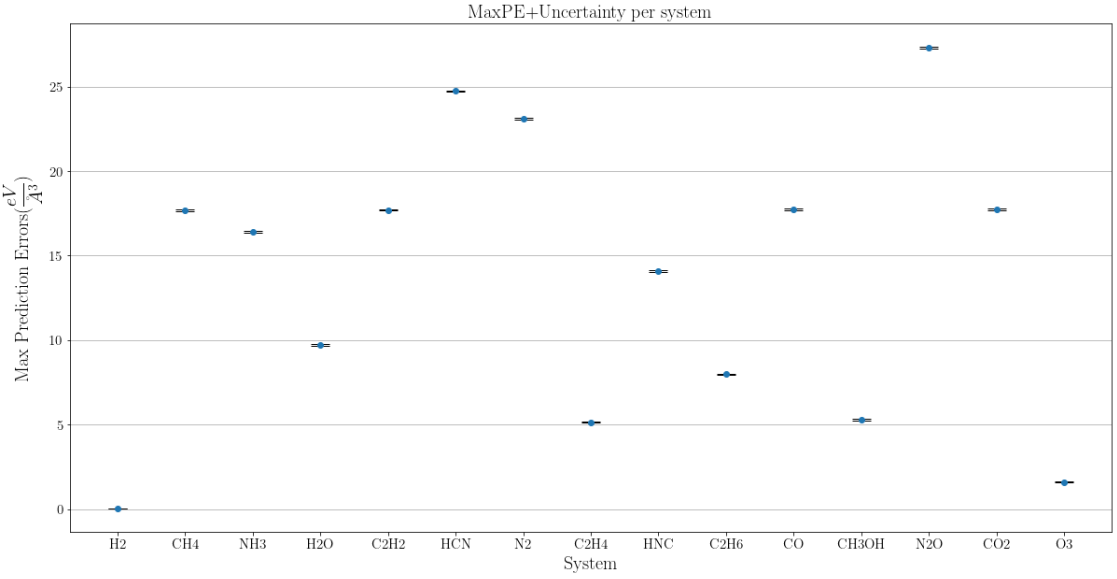
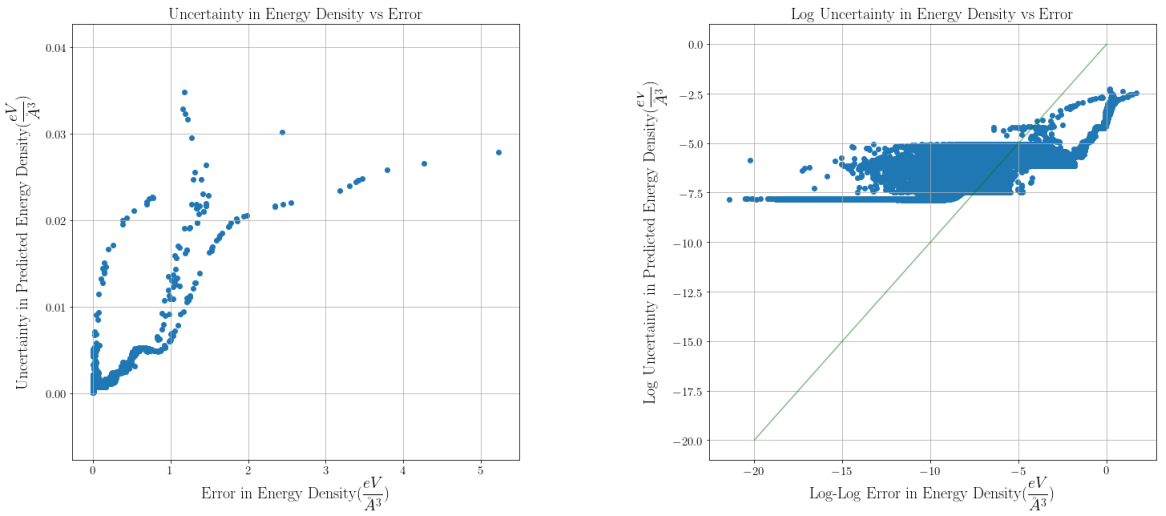


Figure 4.7: Max Prediction Error per system

Fig 4.7 shows the MaxPE per system. The maximum errors likely correspond to high energy core regions. There is significant variation in MaxPE across systems and departure from the trends observed in SAE and MPE. For example, O_3 has relatively higher SAE and MPE than other simpler systems as expected but a very low MaxPE. This suggests that MaxPE is not a good metric to quantify the results from the UQ model.

4.2.2 Uncertainty Quantification



(a) 3σ Uncertainty vs ε_{pred}

(b) Log-Log 3σ Uncertainty vs ε_{pred}

Figure 4.8: ϵ_{pred}vs ρ and $\epsilon_{pred-NN}$ vs $\epsilon_{pred-VWN}$

Fig. 4.8 shows the uncertainty in prediction vs error in prediction ε_{pred}. The points above the identity line ($\varepsilon_{pred} > 3\sigma$) are high energy core regions that cannot be predicted correctly using only local information. From Fig 4.8 (b), the majority of points are within tolerance.

Table 4.3: Percentage of points with uncertainty error within uncertainty $n\sigma$

System	σ	2σ	3σ
H2	18.62	99.85	99.91
CH4	38.77	99.00	99.21
NH3	32.19	98.97	99.34
H2O	23.92	98.75	99.13
C2H2	50.55	98.80	99.07
HCN	37.66	98.68	99.07
N2	29.32	98.47	98.94
C2H4	52.19	98.43	98.73
HNC	40.64	98.73	99.14
C2H6	54.47	98.13	98.51
CO	32.81	98.55	98.98
CH3OH	42.58	97.96	98.46
N2O	32.09	97.43	98.15
CO2	32.36	97.45	98.16
O3	31.69	96.81	97.67
Mean	36.65	98.40	98.83

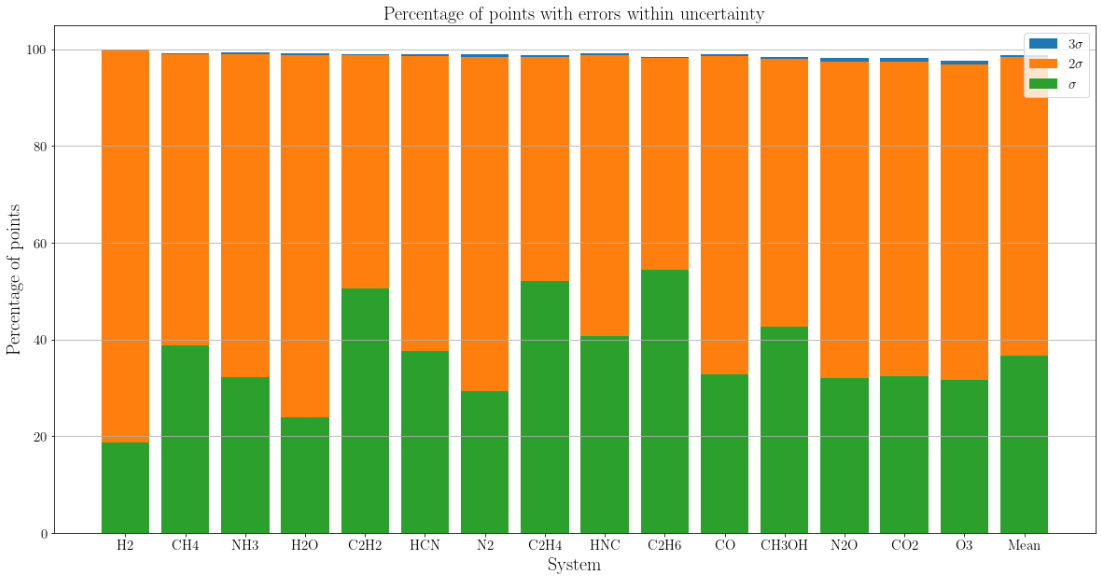


Figure 4.9: Percentage of points with uncertainty error within uncertainty $n\sigma$

For normally distributed errors, it is expected that 65% of points would be within 1σ standard deviation, 96% within 2σ and 99% within 3σ [52]. Across all systems, an average of 36.65% of errors fall within 1σ. This suggests that errors are being significantly underestimated within 1σ and that the error distribution might not be gaussian. However, 98.83% of the points have local errors within 3σ ($\approx 99\%$) tolerance interval (Table 4.3, Fig 4.9).

Effects of Ensemble Size

The number of submodels used impacts the performance of the ensemble system [53]. It is not guaranteed that performance increases monotonically with increasing number of submodels. There exist peak values beyond which adding more submodels leads to diminishing returns or even performance degradation [54]. Uncertainty quantification is not possible for VWN-LDA and single learner systems. As the number of submodels in the ensemble increases, uncertainty increases till ensembles of size 5 and then converges. Ensembles of size 6 and onwards have similar tolerance ranges. Ensembles of size 5 are as effective as ensembles of size 10 at half the computational cost. This can be used to create faster uncertainty quantification systems. Fig 4.10 shows the relationship between SAE and ensemble size for ethane C_2H_6.

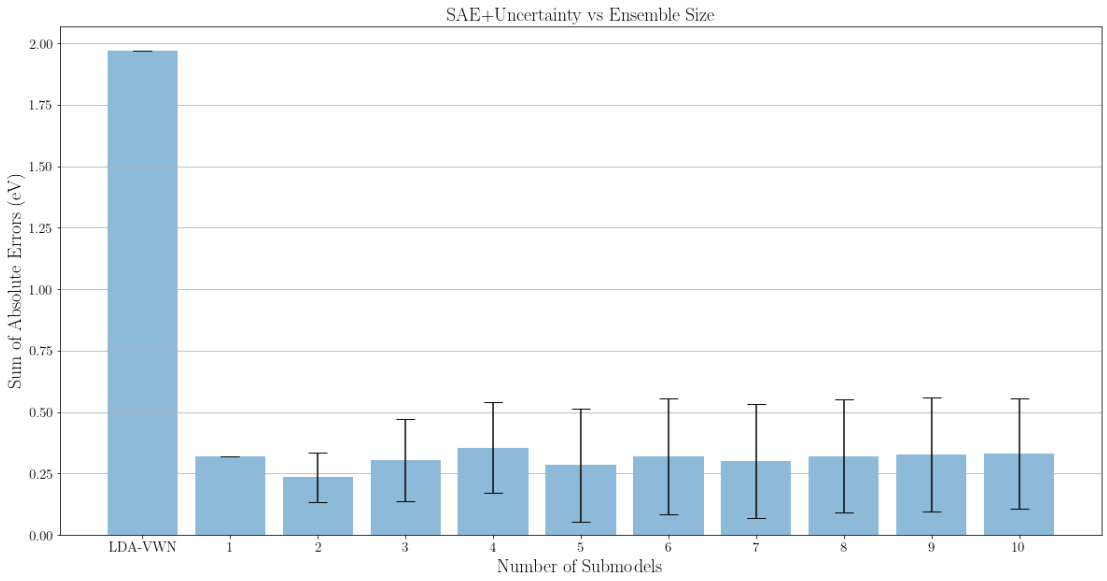


Figure 4.10: SAE and Uncertainty for varying ensemble size, for C2H6

CHAPTER 5

CONCLUSION

We demonstrated the viability of this workflow on a dataset consisting of 15 molecules resulting in a surrogate functional that generalized across all systems. For this demonstration, we used an ensemble for weak learners. While we did not obtain the desired accuracy, we successfully estimated the uncertainty of predictions from our surrogate functional. There are many open design questions that can be explored. In the future, we hope to build on this workflow and achieve chemical accuracy by using an ensemble of more sophisticated learners.

REFERENCES

[1] J. Simons, *An Introduction to Theoretical Chemistry*. Apr. 2003, p. 476.

[2] J. Molenda, "Electronic structure engineering' in the development of materials for Li-ion and Na-ion batteries," *Advances in Natural Sciences: Nanoscience and Nanotechnology*, vol. 8, no. 1, p. 015 007, 2017.

[3] I. Baraffe, G. Chabrier, J. Fortney, and C. Sotin, "Planetary internal structures," 2014. arXiv: 1401.4738.

[4] P. Hohenberg and W. Kohn, "Inhomogeneous Electron Gas," *Physical Review*, vol. 136, no. 3B, B864–B871, 1964.

[5] W. Kohn and L. J. Sham, "Self-Consistent Equations Including Exchange and Correlation Effects," *Physical Review*, vol. 140, no. 4A, A1133–A1138, 1965.

[6] T. Mueller, A. G. Kusne, and R. Ramprasad, "Machine Learning in Materials Science," in, John Wiley & Sons, Ltd, 2016, pp. 186–273.

[7] R. O. Jones, "Density functional theory: Its origins, rise to prominence, and future," 2015.

[8] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," *Phys. Rev*, vol. 10, no. 1103, A1133–A1138, 1965.

[9] A. D. Becke, "Correlation energy of an inhomogeneous electron gas: A coordinate space model," *The Journal of Chemical Physics*, vol. 1062, no. 1988, pp. 21–9606,

[10] J. P. Perdew and W. Yue, "Accurate and simple density functional for the electronic exchange energy: Generalized gradient approximation," *Physical Review B*, vol. 33, pp. 163–1829, 1986.

[11] J. P. Perdew, "Jacob's ladder of density functional approximations for the exchange-correlation energy," *In*, vol. 10, p. 1063, 2001.

[12] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, "Bypassing the Kohn-Sham equations with machine learning," *Nature Communications*, vol. 8, no. 1, p. 872, 2017.

[13] M. G. Medvedev, I. S. Bushmarinov, J. Sun, J. P. Perdew, and K. A. Lyssenko, "Density functional theory is straying from the path toward the exact functional," *Science*, vol. 10, no. 1126, pp. 49–52, 2017.

[14] J. D. Whitfield, N. Schuch, and F. Verstraete, "The computational complexity of density functional theory," in *Many-Electron Approaches in Physics, Chemistry and Mathematics: A Multidisciplinary View*, V. Bach and L. Delle Site, Eds. Cham: Springer International Publishing, 2014, pp. 245–260, ISBN: 978-3-319-06379-9.

[15] K. Burke, "Perspective on density functional theory," *The Journal of Chemical Physics*, vol. 136, no. 15, p. 150 901, 2012.

[16] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biology*, vol. 52, pp. 92–8240, 1990.

[17] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 366, no. 1989, pp. 893–6080,

[18] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251 –257, 1991.

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, p. 436, 2015.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[21] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (Almost) from Scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[22] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, and T. Sainath, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, vol. 29, 2012.

[23] M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, "Deep learning of the tissue-regulated splicing code," *Bioinformatics*, vol. 30, no. 12, pp. i121–i129, 2014.

[24] N. Perraudin, M. Defferrard, T. Kacprzak, and R. Sgier, "DeepSphere: Efficient spherical Convolutional Neural Network with HEALPix sampling for cosmological applications," 2018. arXiv: 1810.12186.

[25] L. P. Levasseur, Y. D. Hezaveh, and R. H. Wechsler, "Uncertainties in Parameters Estimated with Neural Networks: Application to Strong Gravitational Lensing," 2017. arXiv: 1708.08843.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[27] D. Warde-Farley, I. J. Goodfellow, A. Courville, and Y. Bengio, "An empirical analysis of dropout in piecewise linear networks," 2013. arXiv: 1312.6197.

[28] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, 1st. Chapman & Hall/CRC, 2012, ISBN: 1439830037, 9781439830031.

[29] P. Sollich and A. Krogh, "Learning with ensembles: How overfitting can be useful," in *Advances in Neural Information Processing Systems 8*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds., MIT Press, 1996, pp. 190–196.

[30] M. Rupp, "Machine learning for quantum mechanics in a nutshell," *International Journal of Quantum Chemistry*, vol. 115, no. 16, pp. 1058–1073, 2015.

[31] K Mills, M Spanner, and I Tamblyn, "Deep learning and the Schrödinger equation," 2017.

[32] R. M. Balabin and E. I. Lomakina, "Neural network approach to quantum-chemistry data: Accurate prediction of density functional theory energies," *The Journal of Chemical Physics*, vol. 131, no. 7, p. 074 104, 2009.

[33] A. A. Peterson, R. Christensen, and A. Khorshidi, "Addressing uncertainty in atomistic machine learning," *Physical Chemistry Chemical Physics*, vol. 19, no. 18, pp. 10 978–10 985, 2017.

[34] L. Breiman, *Bagging predictors*, 1994.

[35] A. D. Becke, "Densityfunctional thermochemistry. III. The role of exact exchange," *The Journal of Chemical Physics*, vol. 98, no. 7, pp. 5648–5652, 1993.

[36] R. D. Johnson, *NIST computational chemistry comparison and benchmark database*. August, 2011.

[37] X. Lei and A. J. Medford, "Design and Analysis of Machine Learning Exchange-Correlation Functionals via Rotationally Invariant Convolutional Descriptors," 2019. arXiv: 1901.10822.

[38] *Fourier transforms (scipy.fftpack)*.

[39] S. H. Vosko, L. Wilk, and M. Nusair, "Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis," *Canadian Journal of Physics*, vol. 58, p. 1200, Aug. 1980.

[40] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," Tech. Rep.

[41] S. S. Haykin, *Neural networks : a comprehensive foundation*. Prentice Hall, 1999, p. 842, ISBN: 0132733501.

[42] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014. arXiv: 1412.6980.

[43] J. A. Aslam, R. A. Popa, and R. L. Rivest, "On estimating the size and confidence of a statistical audit," in *EVT*, 2007.

[44] J. M. Turney, A. C. Simmonett, R. M. Parrish, E. G. Hohenstein, F. A. Evangelista, J. T. Fermann, B. J. Mintz, L. A. Burns, J. J. Wilke, M. L. Abrams, N. J. Russ, M. L. Leininger, C. L. Janssen, E. T. Seidl, W. D. Allen, H. F. Schaefer, R. A. King, E. F. Valeev, C. D. Sherrill, and T. D. Crawford, "Psi4: An open-source ab initio electronic structure program," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 2, no. 4, pp. 556–565, eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.93.

[45] F. Chollet *et al.*, *Keras*, https://keras.io, 2015.

[46] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015.

[47] *PACE — Georgia Institute of Technology — Atlanta, GA*.

[48] H. W. Lin and M. Tegmark, "Criticality in Formal Languages and Statistical Physics," 2016. arXiv: 1606.06737.

[49] S. H. Vosko and L. Wilk, "Influence of an improved local-spin-density correlation-energy functional on the cohesive energy of alkali metals," *Physical Review B*, vol. 22, no. 8, pp. 3812–3815, 1980.

[50] L. A.L. A. Girifalco, *Statistical mechanics of solids*. Oxford University Press, 2003, p. 519, ISBN: 9780195167177.

[51] Azme Khamis, "The Effects of Outliers Data on Neural Network Performance," *Journal of Applied Sciences*, pp. 1394–1398, 2001.

[52] L. Isserlis, "On the value of a mean as calculated from a sample," *Journal of the Royal Statistical Society*, vol. 81, no. 1, pp. 75–81, 1918.

[53] D. Opitz and R. Maclin, "Popular Ensemble Methods: An Empirical Study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

[54] H. R. Bonab and F. Can, "A Theoretical Framework on the Ideal Number of Classifiers for Online Ensembles in Data Streams," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management - CIKM '16*, New York, New York, USA: ACM Press, 2016, pp. 2053–2056, ISBN: 9781450340731.